arXiv:2304.06222v1 [cs.CR] 13 Apr 2023

当前 NIST 轻量级密码标准的实现、攻击与对策：全面综述

JASMIN KAUR，美国南佛罗里达大学ALVARO CINTAS CANTO，
美国玛丽蒙特大学MEHRAN MOZAFFARI KERMANI，美国南佛罗里
达大学REZA AZARDERAKHSH，美国佛罗里达大西洋大学

本综述是首篇关于当前轻量级密码学标准的研究，该标准于2023年制定。轻量级密码学在保障资源受限的嵌入式系统安全方面发挥着至关重要的作用，例如深度嵌入式系统（植入式和可穿戴医疗设备、智能织物、智能家居等）、射频识别（RFID）标签、传感器网络以及对隐私有严格要求的使用场景。美国国家标准与技术研究院（NIST）启动了轻量级密码学标准化进程。经过多年相对漫长的努力，该竞赛最终于2023年2月结束，ASCON胜出。这项轻量级密码标准将用于深度嵌入式架构，通过提供机密性和完整性／身份验证来保障安全（这与传统的 AES-GCM 分组密码相对应；AES-GCM 是 NIST 对称密钥密码标准）。ASCON的轻量级设计采用 320 位置换，并将其位切片为五个 64 位寄存器字，可提供 128 位级别的安全性。本文根据面积、功耗、吞吐量、能量和效率开销，总结了 ASCON 在现场可编程门阵列（FPGA）和 ASIC硬件平台上的不同实现。本文还从代数、立方体／类立方体、伪造、故障注入和功耗分析攻击等方面，回顾了针对 ASCON密码套件各变体实施的多种差分攻击和侧信道分析（SCAs），并介绍了相应的对策。我们还在综述中分享了见解与展望，为不同领域提出新的未来研究方向。本综述是同类研究中的首篇，也是进一步审视 2023年推出的 NIST轻量级密码标准优势与未来方向的重要一步。

CCS 概念：• **硬件** → 专用集成电路；硬件可靠性筛查 .

附加关键词和短语：ASCON、ASIC、差分密码分析、现场可编程门阵列（FPGA）、轻量级密码学（LWC）、机器学习（ML）攻击、NIST、侧信道分析攻击（SCA）。

ACM 引用格式：
Jasmin Kaur、Alvaro Cintas Canto、Mehran Mozaffari Kermani 和 Reza Azarderakhsh。2023年。《当前 NIST 轻量级密码学标准的实现、攻击与对策综合调查》。*ACM*计算调查-，-，第 - 期（2023年4月），16 页。
https://doi.org/10.1145/nnnnnnn.nnnnnnn

作者地址：Jasmin Kaur，南佛罗里达大学，美国佛罗里达州坦帕市，邮编 33620，jasmink1@usf.edu；Alvaro Cintas Canto，玛丽蒙特大学，美国弗吉尼亚州阿灵顿市，邮编 22207，acintas@marymount.edu；Mehran Mozaffari Kermani，南佛罗里达大学，美国佛罗里达州坦帕市，邮编 33620，mehran2@usf.edu；Reza Azarderakhsh，佛罗里达大西洋大学，美国佛罗里达州博卡拉顿市，邮编 33431，razarderakhsh@fau.edu。

特此授权免费制作本作品的全部或部分内容的数字或纸质副本，供个人或课堂使用，但不得为营利或商业利益而制作或传播副本，且副本必须包含本声明及首页上的完整引文。对于本作品中由 ACM 以外方所有的组件，必须尊重其版权。注明出处后允许摘录。以其他方式复制、重新出版、发布到服务器或向列表重新分发，均须事先获得具体许可和/或支付费用。请通过 permissions@acm.org 申请许可。© 2023年 美国计算机协会。
0360-0300/2023/04-ART- $15.00https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 引言

如今，随着全球广泛采用物联网（IoT）和纳米物联网，轻量级密码学（LWC）已成为必需。LWC 广泛应用于资源受限设备中，例如射频识别（RFID）标签、无线传感器网络（WSN）和嵌入式系统（植入式和可穿戴医疗设备、智能织物、智能家居等），以确保其应用安全。然而，安全并不意味着可靠，许多轻量级密码算法（例如 ASCON）容易受到侧信道分析的攻击。在针对轻量级密码学开展任何标准化工作之前，研究人员已对轻量级密码的各个方面进行了大量研究，其中包括作者开展的研究，例如侧信道分析（SCA） [1]-[7]。

美国国家标准与技术研究院（NIST）启动了轻量级密码学标准化进程。经过多年相对漫长的努力，这场竞赛最终于2023年2月落下帷幕，ASCON [8] 在第三轮其他候选方案中脱颖而出，成为获胜者——Elephant [9], GIFT-COFB [10],Grain128-AEAD [11], ISAP [12], Photon-Beetle [13], Romulus [14],Sparkle [15], TinyJambu [16], 和 Xoodyak [17]。该轻量级密码学标准将用于深度嵌入式架构，以提供安全保障。此前，ASCON 也曾被选为 CAESAR 认证加密竞赛的决赛入围方案。

ASCON是一种轻量级密码套件，提供带关联数据的认证加密（AEAD）以及哈希功能。它采用基于双工结构的运行模式 [8]。ASCON的 320 位置换以位切片方式迭代应用替换-置换网络来加密/解密数据。这种 ASCON 置换的位切片实现使其能够扩展到8 位、16 位、32 位和 64 位平台，同时保持轻量级特性。ASCON 针对不同的消息长度有两个变体——ASCON-128 和 ASCON-128a。ASCON-128 使用64 位的消息长度，而 ASCON-128a 使用 128 位的消息长度。ASCON 还有一种名为 ASCON-128pq的后量子安全变体，它与ASCON-128相同，但使用长度为160位的密钥 [8]。需要指出的是，后量子密码学（PQC）指的是在强大量子计算机存在的情况下能够实施的攻击。公钥密码学算法已于 2022年实现标准化；然而，对称密钥密码学面临的问题要少得多，并且可以使用更长的密钥来应对此类威胁。目前 NIST 的后量子密码学获选算法是 CRYSTALS-KYBER [18], CRYSTALS-Dilithium[19], Falcon [20],和 SPHINCS+ [21]。

多年来，针对不同的 ASCON 变体开展了各种密码分析，包括差分密码分析和侧信道分析。Madushan 等人 [22]探讨了 NIST 轻量级密码标准化流程入围方案的各种故障分析。此外，Dobraunig 等人 [23] 通过使用零和区分器，对 ASCON 进行了深入的密码分析，研究密钥恢复攻击、伪造攻击和代数攻击。此外，他们利用 ASCON 较低的代数次数构造了一个零和区分器，即一组输入和输出值，其和在 $\mathbb{F}_2^n$上为零。对于 12 轮 ASCON，该区分器通过针对第 5 轮后的内部状态，能够以 2^{130} 的复杂度将 ASCON 置换与随机置换区分开来。近期针对 ASCON 的密码分析致力于改进 [23] 的工作，并提出新的方法，以确定针对差分、立方体、代数和伪造类密钥恢复攻击的新型区分器。本研究扩展了 [22], 的工作，并总结了针对硬件/软件实现中的 ASCON 开展的新差分密码分析和侧信道分析研究。

本综述回顾的侧信道分析工作还包括针对 ASCON 硬件实现开展的统计故障分析（SFA）、机器学习（ML）以及基于差分功耗分析的密钥恢复攻击。在统计故障分析中，攻击者对注入 ASCON 置换输出的故障进行统计分析，以完全恢复密钥；在机器学习策略中，则采用深度学习和强化/无监督学习技术来提取

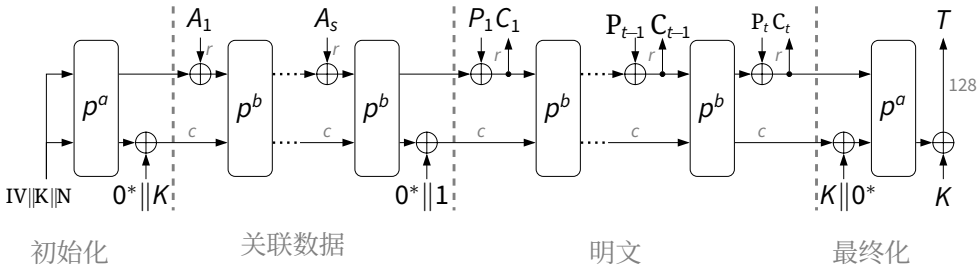


图1. ASCON的关联加密 [8]。

密钥；而差分功耗分析（DPA）则利用ASCON初始化操作中的漏洞，对ASCON硬件实现发起密钥恢复攻击。本文还回顾了SCA对策，包括门限实现策略、更强的S盒设计、错误检测机制，以及防止侧信道泄漏的防护架构。本文还总结了ASCON在现场可编程门阵列和专用集成电路硬件平台上的各种硬件实现，这些实现改进了设计，以提升面积利用率、降低功耗、提高吞吐量、能量效率。我们还在综述中分享见解与展望，为不同领域提供新的未来研究方向。本综述是同类研究中的首篇，也是进一步审视2023年推出的NIST轻量级密码标准的优势与未来方向的一步。

本文的组织结构如下。第2节介绍ASCON的设计与架构。第3节探讨以往及当前文献中提出的ASCON硬件实现。第4节列出针对ASCON开展的现有差分密码分析和侧信道分析（SCAs）。最后，我们在第5节总结本综述。

2 预备知识

ASCON 的完整设计规范见 [8]。ASCON 的加密过程（图 1）采用基于海绵的 MonkeyDuplex结构，由四个阶段组成：初始化操作、处理关联数据、处理明文以及最终化操作。这四个阶段通过两个 320 位置换进行更新，即 p^a 和 p^b，其中 a 和 b 分别表示轮数。这两个置换均按位切片为五个

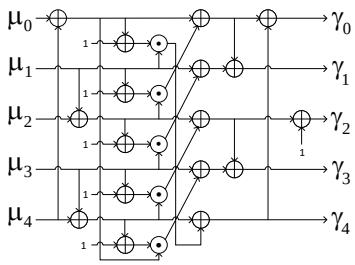


图 2。ASCON 的 5 位替换盒 [8, 53]。

表 1. 输入向量为 μ 时 ASCON-128 非线性 5 位 S 盒 SB 的查找表表示（十六进制形式）

μ	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$SB(\mu)$	4	b	1f	14	1a	15	9	2	1b	5	8	12	1d	3	6	1c
μ	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
$SB(\mu)$	1e	13	7	e	0	d	11	18	10	c	1	19	16	a	f	17

64 位寄存器字，它们构成 5 位内部状态。在完整的 12 轮 ASCON 中，置换会迭代应用基于替换-置换网络（SPN）的轮变换，该变换包括添加轮常数、应用替换层，以及利用线性层对内部状态进行扩散。

替换层由一个非线性 5 位 S 盒（图 2）组成，其十六进制形式见表 1。该 S 盒并行应用 64 次，以更新内部状态的每个位片。5 位替换盒采用布尔逻辑设计，因此在专用集成电路和现场可编程门阵列硬件平台上实现时都非常紧凑、轻量。ASCON 的线性层通过不同的移位值旋转寄存器字，然后对移位后的字值执行模 2 加法，从而更新内部状态中的每个 64 位字。

3 ASCON 的硬件实现

本节介绍 ASCON 系列的各种硬件实现，以及近年来提出的优化方案。FPGA 实现的面积、功耗、延迟和吞吐量开销结果，以及专用集成电路实现的能量利用率开销结果，分别汇总于表 2 和表 3。

ASCON 的各种硬件设计已针对[24] 射频识别标签、无线传感器网络和嵌入式系统等应用进行了实现。这些由 [24] 提出的 ASCON 硬件实现包括：ASCON-fast、ASCON-64-bit 和 ASCON-x-low-area。

ASCON-fast [24] 是一种吞吐量高、处理延迟极低的设计，采用展开的轮变换。该设计不使用流水线，每个时钟周期至少执行一次轮变换，因此可以在单个时钟周期内完成多轮运算。各个 ASCON-fast 变体采用的展开轮变换数量各不相同。展开的轮变换通过少量额外的多路复用器和异或门与数据总线及密钥寄存器相连。

ASCON-64-bit [24] 使用算术逻辑单元（ALU），其控制通路的执行方式与顺序代码类似。除了五个状态寄存器外，该设计还使用两个临时寄存器；这些寄存器与来自控制通路的输入共同构成 ALU 的输入。ALU 接收 64 位数据输入，并使用桶形移位单元、数据存储单元和三种逻辑运算，将数据安排到所选操作数的高位或低位部分。运算结果在 ALU 输出端选出，随后写入目标寄存器。在执行阶段，S 盒和线性层通过 ALU 运算迭代计算，因此每轮运算需要 59 个时钟周期。S 盒的设计经过调整，采用 25 条三操作数指令和两个临时寄存器，以减小面积。

在 [24],提出的 ASCON-x-low-area 变体中，数据通路采用了一种激进的低面积“每周期一位操作”方案。五个状态寄存器均为时钟门控移位寄存器，并具有独立的移位使能输入。在 S 盒计算期间，所有状态寄存器都处于活动状态，数据在每个 S 盒实例中按位切片移位，共经历 64 个时钟周期。线性扩散层通过五个交错的子迭代，分别更新每个状态寄存器。临时移位寄存器用于存储一次迭代中当前线性层的计算结果，随后这些结果

表 2. 开销　　ead 在 FPGA硬件平台上不同 ASCON 硬件实现的结果

ASCON 架构	FPGA 硬件平台	Area (查找表)	功耗 (mW)	吞吐量 (Mbps)	效率 (Mbps/LUT)
ASCON（原始） [53]	Spartan 7	371（切片）	99	6.646	17.914
ASCON（原始版） [53]	Kintex 7	376 (切片)	88	6.709	17.843
ASCON（未防护） [58]	Spartan 6	2048	11.5	255.4	0.1247
ASCON [26]	Artix 7	1808	26.8	39.0	-
ASCON-128 [27]	Artix 7	1330	31	457	0.343
RECO-HCON（128） [29]	Artix 7	1548	-	5926	-
RECO-HCON（128a） [29]	Artix 7	1548	-	9077	-
RECO-HCON（哈希） [29]	Artix 7	1548	-	3160	-
RECO-HCON（hash-a） [29]	Artix 7	1548	-	4534	-
ASCON (受保护) [58]	Spartan 6	6364	37.5	134.6	0.0212
ASCON-128（逻辑单比特） [53]	Spartan 7	373（切片）	99	6705	17.783
ASCON-128（逻辑交错比特） [53]	Spartan 7	380（切片）	99	6687	17.444
ASCON-128（逻辑 CRC-3） [53]	Spartan 7	407（切片）	99	6603	15.601
ASCON-128 (查找表单比特) [53]	Spartan 7	372（切片）	100	6448	17.333
ASCON-128（查找表交错比特） [53]	Spartan 7	377（切片）	100	6443	17.090
ASCON-128（LUT CRC-3） [53]	Spartan 7	425（切片）	100	6431	15.131
ASCON-128（逻辑单比特） [53]	Kintex 7	381（切片）	89	6705	17.598
ASCON-128（逻辑交错比特） [53]	Kintex 7	384（切片）	89	6687	17.414
ASCON-128（逻辑 CRC-3） [53]	Kintex 7	385（切片）	89	6603	17.150
ASCON-128（查找表单比特） [53]	Kintex 7	363（切片）	89	6776	18.667
ASCON-128（查找表交错比特） [53]	Kintex 7	372（切片）	89	6409	17.228
ASCON-128 (LUT CRC-3) [53]	Kintex 7	384（切片）	89	6383	16.224
ASCON-128（展开） [30]	Virtex4	26943	-	817.41	0.031
ASCON-128（递归） [30]	Virtex4	4021	-	506.29	0.125
ASCON-128（展开式） [30]	Virtex7	22636	-	1342.31	0.059
ASCON-128（递归） [30]	Virtex7	2708	-	721.53	0.266
ASCON-128（展开式） [30]	Spartan6	22636	-	688.83	0.031
ASCON-128（递归） [30]	Spartan6	2781	-	346.50	0.124
ASCON-128a（展开） [30]	Virtex4	30006	-	1496.25	0.049
ASCON-128a（递归） [30]	Virtex4	4215	-	970.25	0.231
ASCON-128a（展开） [30]	Virtex7	25187	-	2419.88	0.096
ASCON-128a（递归） [30]	Virtex7	2916	-	1357.08	0.465
ASCON-128a（展开式） [30]	Spartan6	25187	-	1247.22	0.049
ASCON-128a（递归式） [30]	Spartan6	2918	-	638.44	0.218
故障注入 ASCON [49]	SASEBO-GII	217	7.8	198	-
ASCON 上的密钥旁路硬件木马 [33]	SoC Cyclone V	827	22.4	-	-
ASCON 上的轮数缩减硬件木马 [33]	SoC Cyclone V	771	22.3	-	-

在下一次迭代中写回。这个低面积设计每轮变换需要 512 个时钟周期。上述各项实现的所有开销结果均列于表 3。

Diehl 等人 [25] 使用侧信道分析灵活开源工作台（FOBOS）实现的测试向量泄漏评估（TVLA），比较了 ASCON 受保护与未受保护的实现抵御一阶差分功耗分析（DPA）的能力。受保护 ASCON 实现的开销结果见表 2。

[26] 提出了 FOBOS 2，这是 FOBOS 的升级优化版本，用于评估在 Xilinx Artix-7 FPGA 开发板上实现的、具备带关联数据的认证加密（AEAD）功能的各种轻量级密码的功耗测量结果和侧信道分析（SCA）抗性。使用 FOBOS 2 得到的功耗、频率、吞吐量和每比特能量结果列于表 2。结果显示，在 NIST 标准化流程候选方案 Spoc、Spook 和 GIFT-COFB 中，ASCON 表现更优：其功耗最低（50 MHz 时为 33.5 mW），且随着频率升高，动态功耗的增幅也最小。ASCON 的每比特能量为 0.86 nJ/bit，静态功耗约为 27 mW。

文献 [27], 提出了一种基于现场可编程门阵列的 ASCON密码应用方案，适用于便携式医疗物联网 (IoMT) 设备；该密码通过

表 3. ASCON 在专用集成电路硬件平台上的不同硬件实现的开销结果

ASCON 架构	面积（kGE）	功耗 (mW)	吞吐量（Mbps）	能量
ASCON-fast（6 轮，含接口） [24]	25.80	0.184	13,218	23 $\mu J/byte$
ASCON-64-bit（含接口） [24]	5.86	0.032	72	1,397 $\mu J/byte$
ASCON-x-low-area（含接口） [24]	3.75	0.015	14	5,706 $\mu J/byte$
ASCON-fast-TI（6 轮，含接口） [24]	125.19	0.830	9,028	104 $\mu J/byte$
ASCON-x-low-TI（含接口） [24]	9.19	0.045	15	17，234 $\mu J/byte$
RECO-HCON (128) [29]	25.1	1.990	5926	0.335 皮焦/位
RECO-HCON (128a) [29]	25.1	1.990	9077	0.219 皮焦/位
RECO-HCON（哈希） [29]	25.1	1.990	3160	0.637 皮焦耳/比特
RECO-HCON（hash-a） [29]	25.1	1.990	4534	0.439 皮焦/比特
ASCON 互补金属氧化物半导体 [31]	10.1529	0.7459	-	478.9 pJ（64 位）
ASCON 互补金属氧化物半导体 [31]	10.1529	0.7459	-	736.2 pJ（256 位）
ASCON 互补金属氧化物半导体 [31]	10.1529	0.7459	-	1079.4 皮焦耳（512 位）
ASCON 互补金属氧化物半导体/自旋转移矩磁阻式随机存取存储器 [31]	10.7155	0.7148	-	427.9 pJ（64 位）
ASCON 互补金属氧化物半导体/自旋转移矩磁阻式随机存取存储器 [31]	10.7155	0.7148	-	556.5 pJ（256 位）
ASCON 互补金属氧化物半导体/自旋转移矩磁阻式随机存取存储器 [31]	10.7155	0.7148	-	728.1 pJ（512 位）
ASCON 密钥旁路硬件木马 [33]	4024（组合逻辑细胞）	22.4	-	-
ASCON 轮数缩减硬件木马 [33]	3971（组合逻辑单元）	22.3	-	-

带关联数据的认证加密功能增强此类设备的安全性。该方案设计了一种基于轮的 ASCON 架构，每个时钟周期计算一轮。为优化面积，该设计利用赛灵思 7 系列现场可编程门阵列开发板的双输出查找表 (LUT6) 特性，实现 ASCON 的 5 位替换盒。使用 LUT6 实现 ASCON 的 5 位替换盒，在现场可编程门阵列硬件实现中仅需三个查找表；与其他 ASCON 硬件实现相比，显著优化了面积。该硬件实现在赛灵思 Artix-7 系列现场可编程门阵列上完成，面积（以查找表计）、吞吐量、频率和效率（吞吐量/面积）的结果如表 2 所示。与文献 [28] 中的 ASCON 架构相比，该 ASCON密码实现方案的面积减少了 35%，效率提高了 56%。

在 [29], a 中，Wei 等人介绍了一种灵活、可重构且节能的 ASCON 加密处理器。该处理器支持六种不同模式：针对不同数据大小的加密、解密和哈希函数。加密处理器由 ASCON 核心、移位寄存器（FIFO）和 I/O 接口组成。首先，数据和文本输入被载入 128 位移位寄存器作为 FIFO；密钥、随机数和目标实例模式则通过 启动 信号进行处理。在带关联数据的认证加密模式下，ASCON 核心会持续工作，直到整个密文达到与输入消息相同的大小；在哈希模式下，则持续工作直至达到 256 位；或者在处理完成后输出标签验证结果。四个移位寄存器使 ASCON 处理器能够灵活适配具有不同块大小的各种物联网系统，因为它们可用于拆分和填充输入，使其与所用默认变体 ASCON-128a 的块大小相匹配。其他 ASCON 变体 ASCON-128 和 ASCON-Hash 的输入大小，则分别通过将 128 位输入拆分为两个 64 位输入，或采用不同的计数技术以充分利用空间来处理。

ASCON 核心 [29] 由两个阶段组成：选择性异或和参数化置换。启动后，它会迭代运行，使置换模块（第2节）持续工作。每经过固定轮数，置换模块便读取一条新消息。在 ASCON 置换的两次迭代之间，下一次迭代的输入通过将当前海绵状态与另一条消息、密钥或常数进行异或来计算。为支持具有任意轮数、可变异或操作数和块大小的多个实例，我们将当前海绵状态分为两部分：由 128 个最高有效位（MSB）组成的头部，以及由 192 个最低有效位（LSB）组成的尾部。所提出架构在现场可编程门阵列和专用集成电路硬件平台上的面积、频率、功耗、吞吐量和效率开销分别列于表 2 和表 3。

Khan 等人 [30] 探究了 ASCON 在面向人工智能（AI）的物联网设备中的硬件性能。研究采用展开和递归策略，在 Virtex-4、Virtex-7 和 Spartan-6 FPGA 系列上实现 ASCON。展开方案旨在实现高吞吐量，而递归方案则有助于降低硬件成本。对于展开架构，加密/解密通过组合电路执行，并以展开方式部署 ASCON 置换，用于初始化、运行和终结阶段。这样能够获得更高的吞吐量，但代价是较大的面积开销。此外，由于置换函数在每个阶段都使用相同的硬件，作者采用递归策略以实现高吞吐量。为在 ASCON 置换函数中成功实现硬件重用，作者提出每个时钟周期计算两轮置换，因此使用 ASCON-128 加密/解密需要总计 24 个时钟周期（而 36 次置换需要 36 个时钟周期）；ASCON-128a 变体则需要 26 个时钟周期（而 40 次置换需要 40 个时钟周期）。所需的额外异或运算带来的面积开销可以忽略不计。展开实现和递归实现的开销结果均已列于表 2。

Roussel 等人提出了一种基于 CMOS/STT-MRAM 的 ASCON 硬件实现，并在 ASIC硬件平台上进行了基准测试 [31]。该实现通过用非易失性触发器替换易失性CMOS触发器，使其能够抵御电源故障，从而保存 ASCON 计算过程中的中间状态，并在启动时恢复这些状态。与仅采用 CMOS 的 ASCON 实现相比，这种混合CMOS/STT-MRAM实现可将能量利用率提高 11% 至 48%，同时面积开销约为 5.5%。这些结果见表 3。

4 ASCON 的差分与侧信道密码分析

ASCON 等采用认证加密的密码算法存在各种结构和数学层面的漏洞，可被利用来从加密/解密过程中获取置换的信息。本节总结了针对 ASCON 的密码分析研究，涵盖代数攻击、立方体/类立方体攻击、差分攻击、故障攻击和功耗攻击。这些攻击大多针对缩减轮数的 ASCON 版本，以成功恢复密钥。

4.1 代数攻击

在 [32]，Luo 等人通过攻击 ASCON 的置换函数，并使用因子图方法实施软侧信道分析攻击（SASCA），成功恢复了完整的 128 位密钥。该攻击针对 ASCON 的软件实现开展。研究人员利用侧信道信息泄漏上的模板匹配技术，构建了内部置换状态的因子图。随后，他们将该因子图输入所提出的置信传播（BP）算法，以恢复密钥。通过在 8 位平台上运行的模拟可以发现，所提出的攻击只需少量泄漏信息轨迹，利用简单的汉明重量泄漏模型即可恢复完整密钥，同时只产生较低的延迟和内存开销。作者指出，可通过使用多变量值模板或机器学习算法改进该攻击。

4.2 立方体攻击

立方体攻击旨在通过操纵一组立方体变量 [33]，逐位恢复分组密码或流密码的密钥。操纵后的立方体变量随后用于生成加密消息，这些消息构成一个可求解的线性方程组，从而得到密钥位。对于未能通过求解线性方程获得的其余密钥位，可以

通过暴力破解方法获得 [33]。针对减轮 ASCON 变体，人们采用了多种立方体攻击策略和实现方式。最成功的攻击已攻破 ASCON 的 12 轮中的 7 轮 [37]。本节简要汇总针对 ASCON 软件／硬件实现所开展的所有立方体攻击。

Halak和 Duarte-Sanchez [33] 通过在硬件木马（HT）受攻击环境中对 ASCON 置换的初始化函数实施立方体攻击，对轮数减少的 ASCON 发起攻击。他们利用SoC 现场可编程门阵列硬件实现 ASCON-128/128a变体中的漏洞，例如 FSM操作中的未使用/部分使用的状态、初始输入中的未使用值，以及在迭代实现中操纵轮数，以便将 HT 注入硬件。作者提出了两种用于获取轮数减少的 ASCON密钥的HT 设计，即密钥旁路木马和轮数减少木马 [33]。密钥旁路木马在 FSM 中插入恶意状态，操纵控制信号，将密钥直接旁路至密码的输出。轮数减少木马通过在 ASCON加密的初始- 化阶段将置换轮数从12减少到 5，使密码易受立方体攻击，将恢复密钥的时间复杂度从 $2^{103.9}$ 降低至 2^{24} 。HT 受攻击 ASCON实现的开销结果如表3 所示。与原始设计结果相比，产生的面积开销约为 7% [33]。由于注入木马的开销很低，它们在硬件实- 现中不会被检测到。作者提出了多种对策，例如预硅电路验证技术（UCI [34], VeriTrust [35], 或 FANCI [36]），用于检测未使用的输入；后硅功能/结构验证策略；或运行时木马检测技术。还提出了流水线/展开式实现、更好的密钥管理以及 FSM 的独热编码等硬件修改，以增强 ASCON 架构抵御 HT 的能力。

在 [37], Li 等人针对简化的 5/6 轮 ASCON 发起了一次实用的条件立方体攻击。该方法利用一种类似立方体的密钥子集技术 [37]，将其作为密钥划分策略，以针对特定密钥条件恢复整个密钥空间。攻击的实用性在软件环境中进行了测试。然而，完整的 12 轮 ASCON 实现能够抵御此攻击。该攻击之所以成功，是因为密钥划分策略将密钥划分为 63 个密钥子集，从而构造出 65 维立方体。为了确定哪个子集包含正确的密钥位，会分析每个立方体的立方体和 。这一过程会持续进行，直到 63 个密钥子集均通过或未通过立方体测试，从而以 $2^{103.9}$ 的时间复杂度恢复 7 轮 ASCON 的密钥。如果密钥较弱，所提出攻击的时间复杂度还可进一步降低至 2^{77} 。

与 [37], 的研究类似，Baudrin 等人 [38]研究了随机数误用场景下针对 ASCON 的实用条件立方体攻击。该攻击模型旨在通过多次重复使用密钥-随机数对，在加密操作之前恢复 6 轮 ASCON 状态的容量（即未知的内部部分），进而恢复完整状态，并从对应的密文中获取明文信息。在 [38], 中，研究者提出了一种在 ASCON 中搜索条件立方体的新策略：根据经过两轮后二次单项式的系数对公开变量进行划分。这使得攻击者能够恢复 64 至 128 位的内部状态，进而通过暴力破解恢复状态中剩余的 256 位，时间复杂度最高可达 2^{40}。然而，该攻击并未破坏原始的遵循随机数使用规范的 ASCON 设计 [8] 的安全性，其目的仅在于深入了解该密码可能存在的漏洞。

[39] 的作者还研究了在 nonce 误用场景下对 ASCON 变体 ASCON-128 和 ASCON-128p 进行条件立方体攻击。作者提出了一种部分状态恢复方法，并据此实施条件立方体攻击，通过误用 nonce，以 $2^{44.8}$ 的数据复杂度恢复 320 位 ASCON 置换中的 192 位。其余置换比特则通过

在 2^{128} 时间内进行暴力破解来恢复。恢复所有置换比特后，还可使用 $2^{31.5}$ 位数据和 $2^{31.5}$ 位内存，以 $2^{129.5}$ 的时间复杂度恢复密钥。

提出了一种利用深度学习（DL）的基于机器学习（机器学习）的已知明文攻击，用于ASCON的加密操作 [40]。该攻击在随机数误用场景且不存在终结函数的情况下，以 99。8% 的准确- 率预测明文。所提出的攻击无法应用于 ASCON 的实际场景，因为在该场景中会遵守随机数的使用规范，并且在密码函数中引入了足够的随机性，使得基于机器学习的攻击无法有意义地匹配输入和输出。

在 [41], Rohit 等人使用一种名为部分多项式乘法的超多项式恢复技术，对 7 轮 ASCON 的软件实现发起密钥恢复攻击。使用该技术，只需 2^{64} 位数据和 2^{123},的时间复杂度，即可恢复整个 128 位密钥，且仅使用 2^{101} 位内存。利用除法性质，他们还为 7 轮 ASCON 识别出 [42] 个新的立方区分器，并改进了其他减轮 ASCON 实现的立方区分器，从而在随机数/密钥遵循场景下恢复密钥。

4.3 差分密码分析

在 [43], Zong 等人提出了两种针对 ASCON 海绵结构哈希变体（即 ASCON-Hash 和 ASCON-Xof）的基于差分的碰撞攻击策略。第一种攻击不具实用性，针对 2 轮 ASCON-Hash，时间复杂度为 2^{125}。此外，他们还对 2 轮 ASCON-Xof 实施了一种实用的碰撞攻击；当输出长度为 64 位时，时间复杂度为 2^{15} 。研究者采用混合整数线性规划方法和目标差分算法[44]，找出了这些哈希变体的差分特征。

Gerault 等人 [45] 利用约束编程（CP），通过软件对 ASCON 变体 ASCON-128 和 ASCON-128a 的置换进行差分密码分析。他们利用 CP 寻找优良差分的能力，为 ASCON 生成差分特征，以构造受限生日区分器（针对 4 至 7 轮）和矩形攻击（针对 4 轮和 5 轮）。根据这些区分器能否用于攻击带密钥或不带密钥的置换，分别将其分为黑盒和非黑盒类型。此外，他们还提出了高概率差分，以改进针对 ASCON-Hash 的碰撞攻击（在 [43] 中提出），其时间复杂度为 2^{103}。此外，在随机数遵循场景下，研究者还利用多个针对 3 轮和 4 轮的差分特征，对减轮 ASCON-128 和 ASCON-128a 的置换及终结函数实施伪造攻击。对于终结操作的三轮，这类攻击的时间复杂度确定为 $2^{32.76}$。对于 4 轮 ASCON-128，在终结阶段实施这类攻击的时间复杂度为 $2^{96.61}$。然而，在这种情况下，攻击所处理的数据块数量超过了单个密钥所建议的上限。对于 ASCON-128a，伪造攻击同时针对迭代置换和终结函数。对于迭代置换，所提出攻击的时间复杂度为 2^{117}, ，但其处理的数据块数量也超过了 ASCON 设计者为给定密钥设定的上限 [8]。

利用 ASCON S 盒中不受干扰的位，Tezcan [46] 对缩减至 4/5 轮的 ASCON 实施了截断差分、不可能差分和不大可能差分分析。研究发现，在 S 盒的执行过程中存在 35 个不受干扰的位，可用于生成上述差分。不可能差分分析基于这样一个思想：对于特定差分，该差分不会出现，即其概率为零。通过使用不可能差分确定并排除错误密钥，可以帮助找到正确密钥。该密钥恢复攻击使用概率为 1 的截断差分，针对 4/5 轮 ASCON 实施。概率为 1 的截断差分与密码对称性相结合，有助于确定与活跃 S 盒相关联的两个密钥位是否为 1。因此，对于针对 4 轮 ASCON 实施的截断攻击，16 个密钥位将为 1，其余 48 个密钥

可以使用 3^{48} 次加密进行暴力破解，恢复 4 轮 ASCON 中的密钥位，同时使用 2^2 位数据。针对 5 轮 ASCON 的截断攻击可以恢复 70 个密钥位，其余密钥位则可通过 2^{58} 次加密（针对弱密钥）或使用 2^{109} 位数据进行 $2^{128} - 2^{64}$ 次加密来恢复。针对 5 轮 ASCON 的不可能差分攻击与上述截断差分攻击类似，同样使用 2^{109} 位数据，通过补全输出差分来区分 ASCON 置换与随机置换。中间相遇技术与解密操作中的截断差分相结合，仅使用 2^{256} 位数据即可对 5 轮 ASCON 发起不可能差分攻击。

在 [47], Hu 等人对 ASCON 的软件实现进行了概率型和确定性高阶差分/差分线性（HD/HDL）攻击。概率型 HD/HDL 攻击采用高阶代数转换形式函数（HATF）技术，该技术有助于对二次轮函数进行密码分析。利用 HATF，研究人员发现了多种对减轮 5 轮 ASCON 初始化函数的高偏差 $2nd$ 阶 HDL 近似，最高达八阶。因此，利用 HATF，可对 5 轮 ASCON 实施密钥恢复攻击，复杂度为 $2^{22},$；对减轮 ASCON 实施区分攻击，复杂度为 2^{12}。此外，还为 6 轮 ASCON 的初始化函数提出了一种条件 $3rd$ 阶 HDL 近似。使用 HATF 并施加 24 个条件后，在 ASCON 第 6 轮的 S盒中观察到高偏差比特，其理论偏差值为 2^{-22} [47]。该值根据第 5 轮 S盒比特中观察到的偏差（2^{-14}）、累积引理以及应用于 2^{30} 个测试样本的 8 个 HDL 近似计算得出。确定性 HD 攻击还采用了差分支持函数（DSF）技术，该技术通过对置换输入进行高效线性化，有助于找到 HD 区分器。因此，DSF 技术将针对减轮 8 轮 ASCON 置换的区分攻击复杂度从 2^{130} 降至 2^{46}。使用类似的 DSF 方法，可计算出 12 轮 ASCON 置换的零和区分器，其时间复杂度和数据复杂度分别为 $2^{55},$。[47]。

4.4 故障分析

Ramezanpour 等人 [48] 成功地在软件实现中，利用双故障注入和密钥划分技术对 ASCON 的 S盒实施统计无效故障分析（SIFA），从而恢复密钥。作者在 ASCON 终结阶段最后一轮执行的每次加密中，对任意选定的一对 S盒注入故障。故障通过时钟毛刺和/或电压毛刺注入，且注入方式不会影响 S盒的结果。随后，分析由诱导的无效故障产生的正确标签值，以收集有关密钥的信息。在所提出的攻击中，假设概率分布存在偏差；只要有足够的数据可供分析，攻击就能成功（在所提出的研究中，需要 12.5 至 2500 个正确的标签值）。因此，基于 SIFA 的故障模型要求低于其他差分故障分析技术，并且还具有噪声容忍性。错误检测和错误随机化技术等故障攻击对策在 SIFA 面前会失效，因为这些技术依赖于故障注入时密码运算产生的错误输出值。针对 SIFA 的最佳对策是能够检测故障注入机制，或使故障分布独立于秘密数据的对策。基于传感器的技术可以检测故障注入机制，但其能够检测的故障机制有限。现场可编程门阵列和专用集成电路硬件实现被提议作为未来工作。

Surya 等人 [49] 实施了一种同步时钟毛刺策略，以在 ASCON 架构的选定部分诱发延迟故障。硬件实现采用 SASEBO-GII FPGA 开发板，并使用数字时钟管理器（DCM）生成同步时钟信号。

在每个加密轮中，通过高频故障时钟信号将故障注入 ASCON S盒，从而导致 ASCON 线性层产生错误输出。Surya 等人还尝试实现异步时钟毛刺策略，但与同步方法相比，该策略会带来更高的资源开销 [49]。故障注入通过仅向设计的少数部分馈送故障时钟信号来执行，以便更好地观察错误分布和传播，并轻松模拟电磁注入。对此类攻击的可能对策包括采用阈值实现（TI）方案和统一掩码方法来保护 ASCON 架构。硬件实现开销见表 2。

Joshi 和 Mazumdar[50] 通过攻击 ASCON-128 软件实现中 S 盒的漏洞，对其进行了子集故障分析（SSFA）。该策略旨在确定当输入比特为 0 时哪些输出比特会变为 0，从而找出 ASCON S 盒输入比特与输出比特之间的相关性。他们还提出了一种密钥划分策略，将密钥恢复的搜索空间缩小至最坏情况下的 2^{64} 。针对 SSFA 提出的对策包括：密钥白化操作之前进行密钥掩码、通过部分解密进行错误检测，或采用经过改进且能够抵御 1 位 SSFA 的新型 S 盒设计。

在 [50], 中，他们还提出了一种名为初步攻击的故障攻击，该攻击针对终结阶段密钥白化函数和标签生成函数中的漏洞。研究表明，这两种攻击均可通过密钥分析方法恢复整个密钥。对于初步攻击策略，可利用异或函数的对合性质，从生成的标签中恢复密钥值。该攻击可通过三种方式对 ASCON 发动：1）向三个选定的 S 盒注入故障，需要注入 374 次才能恢复完整密钥；2）向一个选定的 S 盒注入故障，随后引发指令跳过错误（由另一次故障注入导致），需要注入 256 次故障才能恢复整个密钥；3）在替换层的输出处将一个字寄存器重置为 0，分别通过 128 次（针对 1 位故障）、16 次（针对 1 字节故障）或 2 次（针对 64 位字故障）故障注入，即可恢复整个密钥。

Ramezanpour等人 [51] 提出了另一种名为故障强度图分析（FIMA）的统计故障分析攻击，用于 ASCON的软件实现。该攻击可以恢复 ASCON 的完整128 位密钥。该攻击设计为利用多种特征来恢复密钥，例如故障密文、SIFA诱导的正确密文以及数据相关偏差。它还能够抵御多种对策，例如用于差分故障分析的错误检测技术；在这种情况下，它可以从增加的样本量中收集秘密信息。即使采用感染式对策，在 ASCON的错误随机轮次中注入故障，FIMA 仍可通过 453 个数据样本恢复密钥。因此，与其他故障分析方法相比，FIMA 的能力强大 6 倍。

Ambili 和 Jose [52] 提出了一种升级版 ASCON-128a 设计，利用元胞自动机（CA）的伪随机性增强密码对 SIFA 和 SSFA 的抵抗能力，并通过软件实现进行了数学验证。CA 是一种技术，其中某个细胞会根据其状态和一组预定义规则，在每次迭代时更新其值。在 [52], 中，使用零边界元胞自动机（将最远的细胞邻居设为 0）来保护 ASCON 架构，方法是在 ASCON 置换的伪随机函数中实现该自动机。该方法能够抵御 SIFA 和 SSFA，是因为线性层中引入了随机性，使得无法可靠地求解针对错误比特推导出的异或/线性方程。作者将其工作的硬件实现可行性作为未来研究方向。

Kaur等人 [53] 提出了低成本错误检测机制，作为ASCON硬件实现抵御故障攻击的对策。他们构建了基于奇偶校验、交织奇偶校验和CRC-3的技术，并将其应用于ASCON的5位替换盒，以检测S盒运算期间可能出现、并导致错误输出的自然故障和瞬态故障。文中介绍了两种错误检测实现方式：一种采用布尔逻辑，另一种采用查找

表（查找表）。针对 640，000 次注入故障对所提出的错误检测方案进行错误覆盖率测试，结果显示覆盖率超过 99。99% [53]。在Spartan-7和Kintex-7现场可编程门阵列上，采用这些错误检测机制保护ASCON硬件实现的开销结果表明，两种实现方式的面积开销均增加至最高15%。结果已列于表 2。这些机制旨在检测DFA和SSFA中利用的大多数注入式单比特和多比特故障；然而，检测SIFA可能颇具挑战，因为错误检测是在S盒输出端而非输入端进行的。

4.5 功耗分析

[54] 的研究工作提出了一种基于机器学习（ML）的侧信道分析与强化学习（SCARL）方法，通过使用无监督学习从功耗测量中提取泄漏模型，以获取机密数据。SCARL通过分析ASCON初始化阶段中非线性S盒计算所消耗的功耗，尝试获取密钥。研究人员使用自动编码器处理功耗测量样本，并结合强化学习和演员-评论家网络对功耗特征进行聚类。研究人员使用SCARL攻击Artix-7现场可编程门阵列开发板上ASCON-128的硬件实现，其中使用FOBOS收集64次S盒计算的功耗测量结果。作者成功证明，所提出的SCARL策略能够通过在24,000次加密操作期间获得的功耗测量结果，恢复Artix-7现场可编程门阵列上所实现ASCON-128密码的密钥；使用SCARL可在8分钟内获取密钥的前4位。

基于功耗泄漏的侧信道分析对策评估使用 FOBOS 2 进行，结果如 [26] 所示。使用 Artix-7 和 Spartan-6 现场可编程门阵列开发板，对受保护和未受保护的 ASCON 架构进行测试向量泄漏评估（TVLA），结果如下。与受保护版本相比，未受保护版本存在显著泄漏；受保护版本的数值则在阈值范围内。不过，未通过功耗泄漏恢复出任何机密数据。还使用与 TVLA 相同的测试向量，对受保护和未受保护的 ASCON 架构进行 χ^2-检验，以评估固定频率类和随机频率类下的泄漏情况。与 TVLA 类似，χ^2-检验发现未受保护的 ASCON 存在泄漏，而受保护版本未发现泄漏。

ASCON 的初始化阶段最容易受到功耗分析攻击，因为初始 S 盒的 5 个输入比特中，只有 2 个是未知的（密钥位），其余 3 个是已知的。在[24],中，除了 ASCON 的优化硬件实现外，作者还提出了针对侧信道分析攻击的对策，尤其是一阶差分功耗分析（DPA）攻击。具体而言，作者采用 TI 方案[55]-[56], ，这是一种掩码技术：对关键数据的计算通过称为份额的变换间接完成。所提出的 ASCON 受保护实现高效地采用了三个份额，与 Keccak 中的做法类似，因为 ASCON 使用了 Keccak 的 χ 函数 [56]的仿射变换。ASCON 的线性层可以在每个份额上实现。然而，非线性 S 盒层需要经过变换，以满足以下性质：1）正确性——结果输出份额之和与对输入份额之和应用 S 盒所得的输出一致；2）非完备性——三个 S 盒函数中的每一个都至少独立于一个输入份额；3）均匀性——每个 S 盒函数都可逆。该研究工作 [24] 还实现了 ASCON-fast 和 ASCON-x-low-area 变体的三份额 TI 版本。ASCON-fast-TI 是一种基于微控制器的实现，被设计为密码协处理器，由微控制器执行初始状态共享和随机数生成。ASCON-x-low-TI 变体则在每个周期的 S 盒运算中直接使用可用随机数生成器的输出。TI 保护实现的硬件实现结果列于表 2 中。

在 [57], 分别对并行实现的ASCON-128和TI保护的ASCON-128发起相关功耗分析（CPA）和差分功耗分析攻击。通过攻击初始化阶段末尾的漏洞，成功对ASCON-128实施了CPA攻击，只需较少的功耗迹线即可获取一半的密钥。利用初始化轮中的这些漏洞，再次对TI保护的ASCON-128发起攻击，并使用偏度差实施三阶攻击 [57]。

同样， [58] 的研究通过每 7 个时钟周期执行一轮，为 ASCON 实现了基于TI的保护方案，以抵御一阶差分功耗分析。该方案实例化了一个混合二份/三份TI保护的64位与模块，每个时钟周期使用 192 个随机比特：其中 128 个用于二份/三份之间的重新分享，其余 64 个用于维持 TI均匀性。通过分析 [58]的 *t*检验泄漏检测测试结果，该 TI 保护的 ASCON 实现被证明能够抵御一阶差分功耗分析。

5 我们的见解、展望与结论

本综述是首篇针对当前轻量级密码学标准的研究；该标准于 2023年制定。本研究涵盖了近年来针对 NIST轻量级密码竞赛获胜者ASCON 提出的各种硬件实现，这些实现面向现场可编程门阵列和 ASIC硬件平台。这些硬件实现针对 ASCON 在资源受限设备中的应用，在面积、吞吐量、效率或能量/功耗利用方面对原始设计进行了改进。本文还回顾了在硬件和软件平台上针对 ASCON 开展的差分密码分析与侧信道密码分析。差分密码分析技术揭示了减轮ASCON的置换函数存在的漏洞，但完整的 12 轮 ASCON 并不存在这些漏洞。研究还表明，S盒设计容易受到 SFA 攻击，这是因为其设计使得攻击者可以利用输入比特与输出比特之间的相关性恢复密钥。研究人员还使用机器学习策略或差分功耗分析对 ASCON 发起功耗分析攻击，以从侧信道泄漏中获取密钥信息；不过，采用 TI 的 ASCON 防护架构被认为能够抵御此类攻击。

一个值得探索的思路是，研究如何为 ASCON 实现增加低成本的故障诊断功能，并将其作为设计决策因素。在设计密码算法之前考虑低开销的优点在于，由此产生的 ASCON 架构将针对低开销错误检测进行设计，而不是事后才制定对策。

另一个与故障攻击和侧信道攻击相关的有趣见解/展望是组合攻击与对策。我们认为，这将是针对 ASCON 的此类攻击的未来发展方向（此前相关研究很少，而且没有研究考虑 ASCON [59]-[62]）。

致谢

本研究由美国商务部、美国国家标准与技术研究院（NIST）提供的美国联邦机构资助项目60NANB20D013资助。

参考文献

[1] J. Kaur、M. Mozaffari Kermani 和 R. Azarderakhsh。2022年。具备错误检测机制的轻量级密码分组算法 QARMA 的硬件构造。*IEEE* 新兴计算主题汇刊，10(1)，514-519。[2] J. Kaur、A. Sarker、M. Mozaffari Kermani 和 R. Azarderakhsh。2022年。基于 FPGA 基准测试的轻量级 Welch-Gong（WG）流密码 WAGE 错误检测硬件构造。*IEEE* 新兴计算主题汇刊，10(2)，1208-1215。[3] A. Aghaie、M. Mozaffari Kermani 和 R. Azarderakhsh。2018年。基于 FPGA 基准测试的硬件和软件高效分组密码 KLEIN 的可靠性与故障诊断架构。*IEEE*计算机辅助设计集成电路与系统汇刊，第37卷(4)，901-905。

[4] S. Subramanian、M. Mozaffari Kermani、R. Azarderakhsh 和 M. Nojoumian。2017年。密码分组密码 LED 和 HIGHT 的可靠硬件架构。IEEE计算机辅助设计集成电路与系统汇刊，36(10)，1750-1758。[5] P. Ahir、M. Mozaffari Kermani 和 R. Azarderakhsh。2017年。基于 FPGA 的可靠性与故障检测轻量级架构：用于 Simon 和 Speck 密码算法。ACM嵌入式计算系统汇刊，第16卷(4)，109:1-109:17。[6] A. Aghaie、M. Mozaffari Kermani 和 R. Azarderakhsh。2017年。基于 FPGA 基准测试的低能耗分组密码 Midori 故障诊断方案。IEEE超大规模集成（VLSI）系统汇刊，25(4)，1528-1536。[7] M. Mozaffari Kermani 和 R. Azarderakhsh。2013年。经 ASIC 和 FPGA 基准测试的可靠轻量级密码学 ISO/IEC 标准 CLEFIA 高效故障诊断方案。IEEE工业电子学汇刊，60(12)，5925-5932。[8] C. Dobraunig、M. Eichlseder、F. Mendel 和 M. Schlffer。2021年。ASCON v1.2：轻量级认证加密与哈希。密码学杂志，34，1-42。[9] T. Beyne、Y. L. Chen、C. Dobraunig、B. Mennink。2021年。Elephant v2。1-55。 [在线] 可访问：https://www.esat.kuleuven.be/cosic/elephant。[10] S. Banik、A. Chakraborti、A. Inoue、T. Iwata、K. Minematsu、M. Nandi、T. Peyrin、Y. Sasaki、S. M. Sim 和 Y. Todo。2020年。GIFT-COFB。密码学电子印本档案。732-768。 [在线] 可访问：https://eprint.iacr.org/2020/738。[11] M. Hell、T. Johansson、A. Maximov、W. Meier、J. S¨onnerup 和 H. Yoshida。2021年。Grain-128AEADv2。提交至 NIST 轻量级密码标准化进程的方案，1-38。可访问 [在线]：https://grain-128aead.github.io。[12] C. Dobraunig、M. Eichlseder、S. Mangard、F. Mendel 和 T. Unterluggauer。2017年。ISAP——迈向侧信道安全的认证加密。载于《对称密码学交易论文集》（ToSC），2017(1)，80–105。[13] Z. Bao、A. Chakraborti、N. Datta、J. Guo、M. Nandi、T. Peyrin 和 K. Yasuda。2021年。PHOTON-beetle 认证加密与哈希族。提交至 NIST 轻量级密码标准化进程的方案。1-115。可访问[在线]：https://www.isical.ac.in/~lightweight/beetle。[14] C. Guo、T. Iwata、M. Khairallah、K. Minematsu 和 T. Peyrin。2021年。Romulus v1. 3。提交至 NIST 轻量级密码学标准的方案，1-57。可访问 [在线]：https://romulusae.github.io/romulus。[15] C. Beierle、A. Biryukov、L. Cardoso dos Santos、J. Groschdl、L. Perrin、A. Udovenko、V. Velichkov 和 Q. Wang。2020年。使用 Sparkle置换族实现轻量级 AEAD 与哈希。ToSC期刊。208-261。[16] H. Wu 和 T. Huang。2021年。TinyJAMBU：轻量级认证加密算法族（第2版）。提交至 NIST 轻量级密码标准化进程的方案。1-40。可访问 [在线]：https://csrc.nist.gov/Projects/lightweight-cryptography/finalists。[17] J. Daemen、S. Hoffert、M. Peeters、G. Van Assche 和 R. Van Keer。2020年。Xoodyak——一种轻量级密码方案。ToSC期刊，60-87。[18] R. Avanzi、J. Bos、L. Ducas、E. Kiltz、T. Lepoint、V. Lyubashevsky、J. M. Schanck、P. Schwabe、G. Seiler、D. Stehl。2017年。CRYSTALS-Kyber：一种 CCA安全的模格基KEM。载于 IEEE 欧洲安全与隐私研讨会（EuroS&P）论文集，353-367。[19] L. Ducas、E. Kiltz、T. Lepoint、V. Lyubashevsky、P. Schwabe、G. Seiler 和 D. Stehl。2018年。CRYSTALS-Dilithium：一种基于格的数字签名方案。密码、硬件与嵌入式系统汇刊（TCHES），2018(1)，238–268。[20] P. A. Fouque、J. Hoffstein、P. Kirchner、V. Lyubashevsky、T. Pornin、T. Prest、T. Ricosset、G. Seiler、W. Whyte 和 Z. Zhang。2018年。Falcon：基于快速傅里叶格的 NTRU 紧凑型签名。提交至 NIST 后量子密码标准化进程的方案。36(5)。1-75。[21] D. J. Bernstein、A. Hlsing、S. Klbl、R. Niederhagen、J. Rijneveld 和 Peter Schwabe。2019年。SPHINCS+ 签名框架。载于 ACM SIGSAC 计算机与通信安全会议（CCS '19）论文集，2129–2146。[22] H. Madushan、I. Salam 和 J. Alawatugoda。2022年。NIST轻量级密码学入围方案及其故障分析综述。电子学期刊，11(24)，4199。[23] C. Dobraunig、M. Eichlseder、F. Mendel 和 M. Schlffer。2015年。Ascon 的密码分析。载于 RSA 会议论文集，371–387。[24] H. Gross、E. Wenger、C. Dobraunig 和 C. Ehrenhfer。2017年。Ascon 硬件实现与侧信道评估。微处理器与微系统期刊，52，470–479。[25] W. Diehl、A. Abdulgadir、F. Farahmand、J.-P. Kaps 和 K. Gaj。2018年。针对选定认证密码的差分功耗分析防护成本比较。密码学期刊，2，3:1-26。[26] A. Abdulgadir、W. Diehl 和 J.-P. Kaps。2019年。用于评估轻量级认证密码硬件实现的开源平台。载于国际可重配置计算 FPGA 会议（ReConFig）论文集，1-5。[27] K. Raj 和 S. Bodapati。2022年。面向医疗物联网设备、使用 ASCON密码对医疗数据进行基于 FPGA 的轻量级加密。载于 IEEE 国际智能电子系统研讨会（iSES）论文集，196-201。[28] S. Khan、W.K. Lee 和 S. O. Hwang。2021年。面向物联网应用的可扩展高效认证加密硬件架构。IEEE物联网期刊，8，14：11260-11275。[29] X. Wei、M. El-Hadedy、S. Mosanu、Z. Zhu、W.-M. Hwu 和 X. Guo。2022年。RECO-HCON：面向可信物联网的高吞吐量可重构紧凑型 ASCON 处理器。载于 IEEE 国际片上系统会议（SOCC）论文集，1-6。

[30] S. Khan、W. K. Lee 和 S. O. Hwang。2022年。评估面向人工智能赋能的物联网设备的 Ascon 轻量级认证加密性能。载于 TRON 研讨会论文集（TRONSHOW），1-6。[31] N. Roussel、O. Potin、G. Di Pendina、J. -M. Dutertre 和 J. -B. Rigaud。2022年。基于互补金属氧化物半导体/自旋转移矩磁阻式随机存取存储器的 Ascon 轻量级密码：一种高能效硬件实现。载于 IEEE 国际电子电路系统会议论文集（ICECS），1-4。[32] S. Luo、W. Wu、Y. Li、R. Zhang 和 Z. Liu。2022年。一种针对 Ascon 的高效软分析侧信道攻击。载于国际无线算法、系统与应用会议论文集，389–400。[33] B. Halak 和 J. Duarte-Sanchez。2020年。针对木马感染的 Ascon 硬件实现的立方体攻击。载于 IEEE 国际 SOCC 论文集，43-47。[34] M. Hicks、M. Finnicum、S.T. King、M. M. K. Martin 和 J. M. Smith。2010年。克服不可信计算基：自动检测和移除恶意硬件。载于 IEEE 安全与隐私研讨会论文集，159–172。[35] J. Zhang、Q. Xu、Z. Sun、F. Yuan 和 L. Wei。2013年。VeriTrust：硬件可信性的验证。载于 DAC 论文集，61，1-8。[36] A.Waksman、M. Suozzo 和 S. Sethumadhavan。2013年。FANCI：使用布尔函数分析识别隐蔽恶意逻辑。载于 ACM 计算机与通信安全会议论文集（CCS），697–708。[37] Z. Li、X. Dong 和 X. Wang。2017年。针对减轮ASCON的条件立方体攻击。《密码学杂志》，160-187。[38] J. Baudrin、A. Canteaut 和 L. Perrin。2022年。针对随机数误用的 Ascon 的实用立方体攻击。《对称密码学汇刊》（ToSC），4，120–144。[39] D. Chang、D. Hong 和 J. Kang。2022年。随机数误用场景下针对 Ascon-128 和 Ascon-80pq 的条件立方体攻击。《密码学杂志》，544-567。[40] D. Jankovikj、H. Mihajloska Trpceska 和 V. Dimitrova。2022年。机器学习赋能的减轮ASCON密码分析。载于国际信息学与信息技术会议论文集（CIIT），1-6。[41] R. Rohit、K. Hu、S. Sarkar 和 S. Sun。2021年。针对7轮Ascon的无误用密钥恢复攻击和区分攻击。《密码学电子印本档案》，194-220。[42] Y. Todo。2015年。基于广义积分性质的结构评估。载于国际密码学理论与应用会议论文集（EUROCRYPT 高级密码学），9056，287–314。[43] R. Zong、X. Dong 和 X. Wang。2019年。针对减轮Gimli-Hash/Ascon-Xof/Ascon-Hash的碰撞攻击。《密码学电子印本档案》，1115-1135。[44] I. Dinur、O. Dunkelman 和 A. Shamir。2021年。针对 Keccak-224 和 Keccak-256 的新攻击。载于国际快速软件加密研讨会论文集，7549，442–461。[45] D. Gerault、T. Peyrin 和 Q. Q. Tan。2021年。探索针对 ASCON 的基于差分的区分器和伪造攻击。《密码学电子印本档案》，1103-1138。[46] C. Tezcan。2016年。Ascon 的截断差分、不可能差分和不大可能差分分析。《密码学杂志》，460-468。[47] K. Hu、T. Peyrin、Q. Q. Tan 和 T. Yap。2022年。从代数视角重新审视高阶差分线性攻击。《密码学电子印本档案》，1335-1400。[48] K. Ramezanpour、P. Ampadu 和 W. Diehl。2019年。一种针对 Ascon 认证密码的统计故障分析方法。载于 IEEE 国际硬件安全与可信度研讨会论文集（HOST），41-50。[49] G. Surya、P. Maistri 和 S. Sankaran。2020年。局部时钟毛刺故障注入及其在 ASCON 密码中的应用。载于 IEEE 国际智能电子系统研讨会论文集（iSES），271-276。[50] P. Joshi 和 B. Mazumdar。2021年。SSFA：针对 ASCON-128 认证密码的子集故障分析。《微电子可靠性》，123，114155:1-14。[51] K. Ramezanpour、P. Ampadu 和 W. Diehl。2019年。FIMA：故障强度图分析。载于国际建设性侧信道分析安全设计研讨会论文集（COSADE），11421，63-79。[52] K. N. Ambili 和 J. Jose。2022年。加固轻量级认证加密方案，以抵御统计无效故障攻击。《密码学电子印本档案》，41-59。[53] J. Kaur、M. Mozaffari Kermani 和 R. Azarderakhsh。2022年。用于轻量级认证密码 ASCON 错误检测的硬件构造及其在现场可编程门阵列上的基准测试。IEEE 电路与系统汇刊第二部分：简报，69(4)，2276-2280。[54] K. Ramezanpour、P. Ampadu 和 W. Diehl。2020年。SCARL：针对 Ascon 认证密码的强化学习侧信道分析。1-25。 [在线]。可访问：https://arxiv.org/abs/2006.03995v1。[55] S. Nikova、C. Rechberger 和 V. Rijmen。2006年。抵御侧信道攻击和毛刺的阈值实现。载于国际信息与通信安全会议论文集（ICICS），529-545。[56] B. Bilgin、J. Daemen、V. Nikov、S. Nikova、V. Rijmen 和 G. Van Assche。2014年。高效且抗一阶差分功耗分析的 Keccak 实现。载于智能卡研究与高级应用会议论文集（CARDIS），187-199。[57] N. Samwel 和 J. Daemen。2017年。针对 Ascon 和 Keyak 硬件实现的差分功耗分析。载于计算前沿会议论文集，415-424。[58] W. Diehl、A. Abdulgadir、F. Farahmand、J.-P. Kaps 和 K. Gaj。2018年。选定认证密码抵御差分功耗分析的防护成本比较。《密码学》，2(3)，1-26。[59] F. Regazzoni、T. Eisenbarth、L. Breveglieri、P. Ienne 和 I. Koren。2008年。了解故障攻击对策的存在情况，能否简化针对密码设备的功耗攻击？载于 DFT 论文集，202-210。

[60] F. Regazzoni、T. Eisenbarth、J. GroschŁadl、L. Breveglieri、P.Ienne、I.Koren 和C.Paar。2007 年。带有附加错误检测电路的密码学S盒的功耗攻击抵抗能力。在 *DFT*论文集，508-516。 [61] J. Dofe、H. Pahlevanzadeh和 Q.Yu。2016 年。基于现场可编程门阵列的针对相关功耗分析攻击的抗故障 AES 综合评估。《电子测试期刊》，32(5)，611-624。 [62] H. Pahlevanzadeh、J. Dofe 和 Q. Yu。2016 年。评估采用不同容错机制的 AES的相关功耗分析抵抗能力。在 亚洲及南太平洋设计自动化会议（*ASP-DAC*）论文集，第 661-666 页。